

Amendments to the Claims

1. (Currently Amended) A computer readable medium comprising a data structure comprising computer executable instructions for a method for saving state for a semantically accessible state binding method, ~~the computer readable medium method~~ comprising:

defining a first state frame including a representation of a state of an executing program, comprising values of registers and heap associated with the executing program; and

defining a second state frame including a representation of state changes made by the executing program after the first state frame is created and the second state frame includes a pointer back to the first state frame.

2. (Currently Amended) The computer readable medium of claim 1 further comprising:

defining a third state frame including a representation of state changes made by the executing program after a fork method creates the third frame, and the third state frame includes a pointer back to the second frame.

3. (Currently Amended) The computer readable medium of claim 2 further comprising defining wherein a fourth frame which includes changes made by the executing program after the fork method creates the third state frame and after a set method returns the executing program to the state of the second state frame.

4. (Currently Amended) The computer readable medium of claim 3 further comprising defining a joined state frame including a combination of state changes in the third and fourth frames.

5. (Original) The computer readable medium of claim 3 wherein a first thread of the executing program makes state changes copied in the second frame, and a second thread of the executing program makes state changes copied into the third frame.

6. (Original) The computer readable medium of claim 1 wherein the second state frame includes unchanged state read from the first state frame.

7. (Currently Amended) A computerized method comprising:
receiving via an application programming interface a request to create a state save;
in response to the request, saving a first representation of a state of an executing program
in response to the request comprising copying state of the program required to return to the
moment the state was saved;

maintaining a second representation of subsequent state comprising changes made to the state of the executing program after the first representation; and

resetting the executing program to the saved first representation upon receiving a state set request at the application programming interface.

8. (Currently Amended) A computer system comprising:
memory and a central processing unit executing,

a program including executable instructions and an evolving present state; a state component comprising an initial representation of a prior evolving present state of the program, the initial representation comprising an instruction pointer location,

a subsequent representation of state changes made by the program since the initial representation, and

a method for returning the program state to the prior evolving present state using
the instruction pointer location.

9. (Original) The computer system of claim 8 wherein the state component further includes a method for reading a location and value from the initial representation of the prior evolving present state and storing the value in a subsequent representation of the state changes.

10. (Original) The computer system of claim 8, wherein the state component includes a fork method for maintaining state for a thread spawned by the program and a forked representation of state changes made by the spawned thread of the program.

11. (Original) The system of claim 10 wherein the state component includes a join method for joining state changes made by the forked thread back into state changes of the subsequent representation.

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Canceled)

16. (Canceled)

17. (Original) A computer readable medium comprising computer executable instructions for performing the method of claim 7.

18. (Canceled)

19. (Currently Amended) A computer readable method comprising computer executable instructions for performing a method comprising:

receiving a request from a method, which takes as a parameter a state object, to create a saved state of an executing model;

saving a first representation of a state of the executing model as the state object;

maintaining a second representation of state changes made by the executing model after the first representation; and

reinstating the executing model state to the state of the first representation using the state object.

20. (Canceled)

21. (Canceled)

22. (New) A computer-based process for allowing an executing version of a program to return to an earlier state, comprising:

defining a state component, the state component including:

a method for saving state of the program, the saved program state represented by a state token; and

a method for returning the program to the saved program state using the state token.

23. (New) The method of claim 22 wherein saving state of the program comprises saving heap and register values of the program at the time the state was saved.

24. (New) The method of claim 22 wherein saving state of the program comprises saving a program instruction pointer location at the time the state was saved.

25. (New) The method of claim 22 wherein at runtime the method for saving state of the program is invoked first and second times, and wherein saving state of the program the first time comprises saving heap and register values as a first frame; and wherein saving state of the program the second time comprises saving heap and register values whose values have changed since the saving state of the program the first time.

26. (New) The method of claim 25 wherein the method for returning the program to the saved program state comprises when a variable is accessed, checking the frames in opposite order to their creation until an update for the variable is found.

27. (New) A programming language construct operational to run on a computer system, the programs written in the programming language having an evolving state when executed, comprising:

A state class comprising:

A state object;

A fork method which saves state of the executing program as a first state when invoked with the state object;

A set method which sets the state of the executing program to the first state when invoked with the state object; and

A join method which joins the current state of the executing program with the first state when invoked with the state object.

28. (New) A computer program product comprising a storage medium having embedded thereon computer readable instructions for a process for restoring state of a variable, the process comprising:

Defining a first data structure which comprises a first saved state frame, the first saved state frame comprising heap and resource values necessary to save substantially all of the state of the program at a first given time;

Defining a second data structure which comprises a second saved state frame, the second saved state frame comprising the values that have changed between the first given time and a second given time;

Defining a state class that includes a method to restore state of the program to the second given time by accessing the first state frame and the second state frame.

29. (New) The computer program product of claim 29 wherein the method to restore state comprises following a pointer between the second and first state frames to discover a last updated instance of a variable.